A NEW PARADIGM FOR PERFORMING RISK ASSESSMENT

Judith L. Bramlage
Computer Associates International, Inc.
12120 Sunset Hills Road
Reston, VA 22090
jbramlage@acm.org

When the Office of Management and Budget (OMB) Circular A-130, Management of Federal Information Resources, was updated in 1996, one controversial area was the change from risk analysis to risk management. The intent was to remove any requirements for *paper* exercises and have a meaningful process for employing security protections, but only as much as is really needed. This paper presents a different way of thinking of risk assessment and a tool to support that thinking. It summarizes the prevailing insurance paradigm and describes why it is not effective for government use. It next describes the classic risk assessment model. Finally, we recast the classic risk assessment thinking into a framework of objects and describe the tool that resulted from our work.

Historically, the standard risk assessment process has been focused on identifying the costs associated with risks (e.g., the insurance paradigm). This is a useful method for commercial enterprises, but is incomplete and poorly focused for Federal government use. This method takes the odds that an event will occur multiplied by the loss that would occur if the event transpired, giving what is described in mathematics as the expectation. For example, if the odds of a tornado hitting a specific site were 0.3 in 100 (or 0.3%) and the value of the site to the operation was $200,000, then the expectation is 0.003 x 200,000 = 600. Thus, $600 is the sum that it would be reasonable to pay for insurance or to pay for risk mitigation. The problem is that the result is only as good as the numbers and true value may not be expressible in simple dollars. In fact, the cost in dollars of an operation may not express the criticality of the site. For instance, if the risk is associated with loss of public confidence, it is not adequately expressed in real dollars. The amount of work required to deal with the details of this type of assessment makes it expensive to apply. Moreover, since the assignment of numbers is critical to the process but rarely documented, the foundation of the analysis is seldom obvious.

OMB Circular A-130, Management of Federal Information Resources, recommends applying qualitative, not quantitative, methods to risk management in Federal systems. We have taken the idea of a qualitative risk assessment and developed a method based on the premise that management accepts the residual risk. We also used an object-oriented viewpoint to work logically through the process. This method is supported by an automated tool providing a structured mechanism for performing and documenting a qualitative risk assessment that is criteria-based.

Classic Risk Analysis Process

The classic risk assessment process (Figure 1 and described below) consists of a sequence of steps:

- Identify applicable policies;
- Study current environment;
- Identify requirements;
- Hypothesize vulnerabilities;

- Design and perform tests;
- Identify threats and threat scenarios;
- Correlate vulnerabilities with threats to define risks;
- Evaluate countermeasures; and
- Make recommendations and identify corrective tasks to mitigate risks.

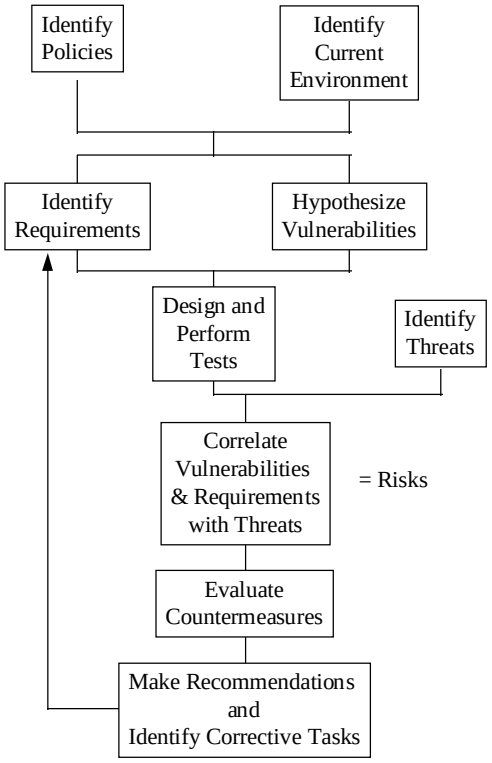

Figure 1 Classic Risk Assessment Model

Identify Applicable Policies: Any determination of risk for a Federal agency automated information system (AIS) starts with examination of the security policy and requirements for the AIS based on Federal laws and regulations and appropriate Department directives. The basic security requirement for Federal systems[1] is Controlled Access Protection[2] with segregation of duties,[3] access by least privilege,[3] and continuity of service.[3]

Study Current Security Environment: A study of the current security environment can assist in determining the existing security countermeasures and pertinent risks. This study includes the security concerns of hardware configuration and functionality, software functionality and integrity, data protection needs, physical facilities and associated security, personnel, and communications operations including message integrity and network availability.

Requirements Analysis: A requirements or sensitivity analysis is performed to determine mission criticality and the sensitivity of the information processed. The information gathered in this sensitivity analysis is used to determine the security impact if data is disclosed or if data integrity is lost. It is also used to estimate the impact of denial of service conditions.

Hypothesized Vulnerabilities: Based on the environment, vulnerabilities are hypothesized. These are possible weaknesses in system security procedures, system design, implementation, etc., that could be exploited through successful perpetration of a threat to violate the formal and informal

---

[1]The White House National Telecommunications and Information Systems Security Committee directs that Federal agencies provide automated Controlled Access Protection (C2 level) for all sensitive or classified information processed or maintained by AIS, when all users do not have the same authorization to use the sensitive information. [NTISSP 200]

[2]As defined in Trusted Computer System Evaluation Criteria, DoD 5200.28STD, Controlled Access Protection (C2)

[3]Office of Management and Budget Circular A-130, Management of Federal Information Resources, Appendix III

security policies. Vulnerabilities include obtaining, modifying, and destroying data, and disrupting operations.

Test Design and Performance: Tests are the measures of the success or failure of requirements and exposures from vulnerabilities that could create risks. Tests include verification of requirements compliance through documentation and observation and checking of vulnerabilities through hands-on testing. In some cases, new vulnerabilities may be discovered during the testing process.

Identification of Threats: A threat is any circumstance or event with the potential to cause harm to a system in the form of disclosure, destruction, or modification of information, or denial of service.

Identification of Risks: Risk is the potential for the perpetration of a threat and the successful exploitation of a vulnerability or unmet requirement. The types of risks are unauthorized disclosure of information, unauthorized modification or destruction of information, and denial of system services. The risk of each vulnerability or unmet requirement being exploited is subjectively determined. In a quantitative risk analysis, the degree of risk (i.e., a combination of the sensitivity of the vulnerable information or communications service and the probability of exploitation) associated with a particular threat/vulnerability scenario is determined by the mathematical expectation (i.e., the risk is determined by the sum of all the chances of a loss times the cost of each loss). If the risk analysis is qualitative, the risk for each loss can be assigned a value, either true/false or high/medium/low.

Evaluation of Countermeasures: This step identifies both the security countermeasures and their effectiveness in mitigating the risks.

Recommendations and Corrective Tasks: The risk assessment is correlated to the security policy to ensure the two are consistent. The security policy establishes the security rules that must be satisfied within the agency perimeter. If a gap exists between the security policy and the design, implementation, or installations then the gap must be closed. However, prior to closing such a gap, a cost-benefit justification needs to be provided to justify the cost versus the reduced risk.

The classic risk analysis process is supported by risk analysis tools. In general, these tools require an inventory, thus force a focus on value and odds or "success probability," making them quantitative. If the mode of thinking about risk assessment is modified to object-oriented, a new viewpoint emerges. That viewpoint resulted in a new tool for our organization.

New Risk Assessment Paradigm

We needed a tool that would allow us to have a standardized method of documenting risks. It had to be repeatable and use previous data from year to year. It should be possible for the information to be collectible by field support personnel who are not expert in security evaluation and risk analysis, and evaluated by security experts to produce the risk assessment. It had to be criteria-based. It had to be self-documenting. It had to allow us to consolidate at various site (i.e., management) levels.

The results had to be easily explainable to management. The documentation had to include the documentation of the technical and management choices upon which the risk assessment is dependent.

We started with the viewpoint that requirements, vulnerabilities, risks, etc., are objects that can be treated logically. Figure 2 identifies the risk assessment objects and the logical relationships between the defined objects.

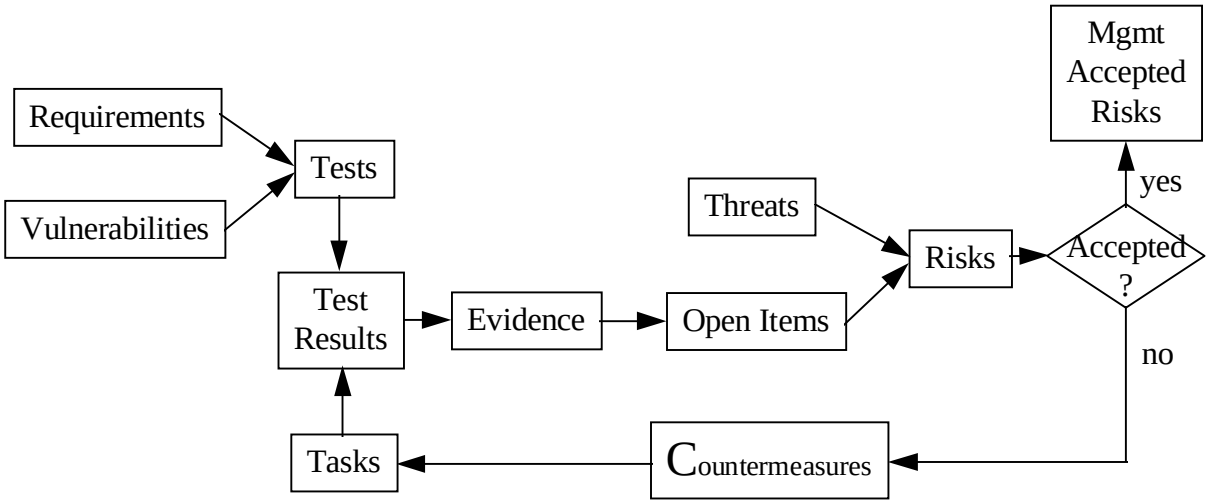

Figure 2. Object-Oriented Risk Assessment Method

This method identifies specific requirements and hypothetical vulnerabilities based on specific criteria. These requirements and vulnerabilities are used to identify tests. The tests are documented in the test results. The test results are re-formed or evaluated into evidence, positive or negative. The vulnerabilities and requirements are correlated against the evidence and the closed and open items are identified. Open items are correlated against the threats, resulting in issues or risks. If the risks are not accepted by management, then the risks and negative evidence are used to create countermeasures, resulting in corrective tasks. Results of the corrective tasks are fed back into the test results and evidence, then the open items and risks are re-evaluated. For the risks accepted by management, the cycle is complete. When all residual risks and identified tasks have been accepted by management, the entire risk assessment is complete. The two areas where judgement is applied are in the management acceptance of risks and in the evidence, where identification is made of whether a requirement or vulnerability is open or closed.

This method utilizes eight definition lists and relates them. The basic tables are Requirements, Vulnerabilities, Tests, Test Results, Evidence, Threats, Countermeasures, and Tasks. The relationships are defined as correlations of the basic tables. The results are lists created from querying the previous tables. Figure 3 provides an integrated example, showing a limited sample of data from the definitions and relationships.

The eight definition lists, described below, are implemented in the risk assessment automated tool as indexed tables containing the name, type, description, and, sometimes, source for that item.. For example, the Requirements table contains the requirement name, the requirement type, the

requirement description, and the source of the requirement. The tables shown in Figure 3 do not exactly correlate to those used in the tool; they are simplified for ease in presenting the concepts and examples.

Requirements: This list describes the requirements and identifies the source of each requirement. For each named requirement, the table provides the Rcode (the requirement table index), the type of requirement, the short name, a full description, and the source. Table 1 shows an extract of the requirements definition table.

Vulnerabilities: This list describes the vulnerabilities and identifies the source of each vulnerability. For each named vulnerability, the table provides the Vcode (the vulnerability table index), the type of vulnerability, the short name, and a full description. Table 2 shows an extract of identified vulnerabilities.

Tests: This list describes the tests and identifies the source of each test. For each named test, the table provides the Test code (the test table index), the type of test, the short name, a full description, and the source. Table 3 shows an extract of identified tests.

Tests Completed: This list identifies the outcome of each specific test. The table contains the index, the Test table index reference, the date the test was accomplished, whether the test is complete, the source of the test, and a memo field for test notes. In the example in Table 4 below, the second test identified in Table 3 has been completed and results obtained.

Evidence: This list contains the evidence collected and identifies the source of each piece of evidence. For each named evidence, the table provides the Ecode (the evidence table index), the type of evidence, the short name, a full description, and the source. Table 5 shows an extract of identified evidence.

Threats: This list contains the threats and identifies the source of each threat. For each named threat, the table provides the Tcode (the threat table index), the type of threat, the short name, and a full description. Table 6 shows an extract of identified threats.

Risks: This list contains the identified risks. For each named risk, the table provides the RKcode (the risk table index), the type of risk, the short name, and a full description. Table 7 shows an extract of identified risks.

Countermeasures: This list describes the countermeasures. For each named countermeasure, the table provides the Ccode (countermeasure table index), the type of countermeasure, the short name, and a full description. Table 8 shows an extract of identified countermeasures.

Tasks: This list describes the tasks for creating the countermeasures. For each named task, the table provides the TKcode (the task index), the type of task, the short name, and a full description. Other information that could be provided includes cost and schedule. Table 9 shows an extract of identified tasks.

The paragraphs below describe the relationships between the definition tables are defined in the relationship tables. Examples correlating to the definition examples are provided. Again, the tables shown do not exactly correlate to those used in the tool. They are simplified for ease in presenting the examples, extracts only are provided, and some tables are combined to reduce the textual redundancy.

Tests to Requirements and Vulnerabilities Tables: These tables describe the relationships between the tests and requirements or vulnerabilities. For each test, the tables identify the requirement (by Rcode, (i.e., index into the requirements table) or vulnerability (by Vcode, i.e, index into the vulnerabilities table). The relationships can be many-to-many. In the example in Tables 10 and 11, the requirements from Table 1 and the vulnerabilities from Table 2 are related to the tests in Table 3.

Test to Evidence Table: This table relates the evidence to tests, and thus to the test completed list which contains the test notes. For each test, the table identifies the appropriate evidence (by Ecode, i.e., the index into the evidence table). In the example in Table 12, the evidence from Table 5 is related to the tests in Table 3.

Requirement/Vulnerability Status Tables: These tables identify whether a given requirement or vulnerability is closed or open based on the evidence. For each named item, the table provides the Rcode (the index to the requirements table) or the Vcode (the index to the vulnerabilities table), the status, and the reason if closed. In the example provided in Table 13, the status list identifies the items in Table 1 and Table 2, the status, the evidence from Table 5 used to close the requirement or vulnerability, and the reason for closure. It can also identify any open items from those tables. [4]

Requirement or Vulnerability/Threat/Risk Correlation Tables: These tables contain the correlations between the requirements or vulnerabilities and threats and resultant risks. The open requirements or vulnerabilities that have a corresponding threat are expressed in terms of the risk. For each correlated requirement or vulnerability and threat to risk, the tables provide the index to the requirement or vulnerabilities table, the index to the threats table, and the index to the risks table. Examples are provided in Tables 14 and 15. In the example in Table 15, the item listed is an open vulnerability from Table 13 where there is a matching threat from Table 6. Since the risk shown in Table 14 is associated with a requirement that is closed, the risk does not propagate to the Risk List. The Risk List is actually an implied list in that it is the result of a query, not maintained in the database.

---

[4]Since security is never absolute, the Requirement/Vulnerability Status Table is the method chosen to add knowledgeable assessment to the risk assessment process. In the example, Table 13, two requirements are closed and the vulnerability is open. The method for completing this table is to display the requirement or vulnerability, to provide on a pull-down menu the associated tests and, on a second pull-down menu, the evidence associated with each test. The user is able to use associate each evidence item to the appropriate vulnerability or requirement and to identify whether it is open or closed, and then to provide the reason if closed.

Accepted Residual Risk Table: This table identifies the items from the Risks List for which management must be willing to accept the risk of *not* applying a countermeasure.[5] The table, illustrated in Table 16, contains the RKcode (the index to the risks list, Table 7), the manager responsible for the decision, and the reason for the decision. In the example, the site Security Officer has accepted the risk, RK017, related to users putting their password on or near their terminal based on the (1) security procedures prohibiting sharing of passwords, (2) the personnel penalty for violations of password sharing, and (3) the training courses that cover this issue.

Risks to Countermeasures Table: This table relates the risks to countermeasures. For each risk, the table contains the RKcode (the index to the Risks table, Table 7) and the Ccode (the index to the countermeasures table, Table 8). In the example in Table 17, there are three countermeasures to the one risk shown. The Countermeasures List is an another implied list based on a query of the Risk List and the Risks to Countermeasures Table.

Countermeasures to Tasks Table: This table relates the countermeasures to the tasks that need to be performed. The table contains the CTcode (the index), the TKcode (the index to the task table, Table 9), and the Ccode (the index to the countermeasures table, Table 8). In the example in Table 18, the countermeasure, C254, is associated with only one task. Again the Task List is the result of a query of the Countermeasures List and the Countermeasures to Tasks Table.

The resultant lists are available from queries:

Open Items are identified from Requirements and Vulnerabilities that have not been closed. They may or may not appear in the status table.

Risk List is identified from Open Items that are correlated to threats and not included in the Accepted Residual Risks table.

Countermeasures List is identified from correlating the Risk List to the related countermeasures.

Tasks List is identified from correlating the Countermeasures List to the related tasks.

Results = Recommended Tasks + Accepted Residual Risk + Accepted Evidence

The results from this risk assessment are directly available from the appropriate tables and lists:

- the recommended tasks identified in Table 18 and described in Table 9
- the accepted residual risks identified in Table 16
- the reasons for acceptance of the evidence described in Table 13.

Using ODBC technology, the information from the tables implemented in an ODBC-compliant database, can be directly linked into word processing text for the certification/accreditation package.

---

[5] Since security is never 100%, this is the method chosen to add management choice to the risk assessment process.

Summary

Since the current tools are tied to quantitative risk assessments, a new method was needed to meet the requirements of the new OMB Circular A-130. We took a logical view and developed an object-oriented tool to simplify our documentation and release us from the need to depend on a complete inventory of the Agency assets. The result was a repeatable method we could apply to a large, distributed AIS where the first set of data would be collected by the local administrators and subsequently coordinated and analyzed by experienced security professionals. The result was a qualitative, repeatable method the Agency can afford to perform. It is suitable for producing individual site reports, summarizing by region, and summarizing as a whole. It supports measurement and tracking from one year to the next. It also feeds directly into the Risk Assessment Report and the AIS certification packages.

References

1. DoD 5200.28 STD, Trusted Computer System Evaluation Criteria (TCSEC), National Computer Security Center, Fort Meade, MD, December 1985.

2. FIPS Publication 31, Guideline for Automated Data Processing Physical and Risk Management, June 1974

3. FIPS Publication 65, Guideline for Automated Data Processing Risk Analysis, August 1979.

4. NTISSP 200. National Policy on Controlled Access Protection. The White House. National Telecommunications and Information Systems Security Committee, July 15, 1987.

5. OMB Circular A-130. Management of Federal Information Resources, Office of Management and Budget, Wash., DC, 1985, 1993, 1994, 1996.

Table 1 Requirement Definitions

Rcode	Type	Requirement	Description	Source
R001	Passwords	Password assignment to users	A password must be initially assigned to a user when enrolled on the system.	CSC-STD-002-85, Password Management Guideline
R002	Passwords	Periodically change passwords	A user's password must be changed periodically.	CSC-STD-002-85, Password Management Guideline

Table 2 Vulnerability Definitions

Vcode	Type	Vulnerability	Description	Source
V001	Password	Written password	Passwords written on note and attached to monitor or other surface near computer.	

Table 3 Test Definitions

TScode	Type	Test	Description	Source
TS101	Password Procedure	Identify password assignment to users	Identify initial password assignment to a new user when enrolled.	Procedures
TS102	Password Parameter	Identify password parameters	Determine current default password parameter assignments	System parameters

Table 4 Tests Completed

TC-Counter	Test	Date	Complete	Source	Notes
TCC102	TS102	8/1/97	X	System parameter table and default user table	PassRqd = True; PassExp=120D

Table 5 Evidence Definitions

Ecode	Type	Evidence	Description	Source
E201	Procedure	Passwords assigned to users	Procedure for password assignment to a new user when enrolled.	User Procedure
E202	Parameter, Password	Password Required = yes	Password parameter for assignment ensures that a password exists	RA Test Report
E203	Parameter, Password	Password Expiration = 120 days	Maximum period for use of a password is 120 days	RA Test Report

Figure 3. Risk Assessment Example (page 1 of 3)

Table 6 Threat Definitions

Tcode	Type	Threat	Description
T10	Human Intentional	Penetration	Penetration involves attacks by unauthorized persons in attempts to gain access to an agency system by defeating its security controls. Penetration is often performed in conjunction with browsing and misuse and results in unauthorized disclosure or modification of information, unauthorized receipt of services, or denial of service to legitimate users or critical functions.
T11	Human Intentional	Misuse	Misuse is the use of processing or communications services for other than official, authorized purposes (e.g., personal gain, espionage). Misuse includes the threats of inadvertent or intentional execution of malicious functions (e.g., virus, worm, Trojan Horse, etc.), performance of undesirable functions (e.g., erasing the file system), and other errors of commission, omission, and oversight. Misuse results in unauthorized disclosure or modification of information, unauthorized receipt of services, or denial of service to legitimate users or critical functions.

Table 7 Risk Definitions

TKcode	Type	Risk	Description
RK124	Hacker	Hacker logon attempt prior to initial password setting	Authorized user's logon is attempted by another user prior to authorized user setting password initially.
RK125	Hacker	Hacker uses password left on note on monitor.	Authorized user writes password on note where others can access it.

Table 8 Countermeasure Definitions

Ccode	Type	Countermeasures	Description
C254	Procedure	Passwords assigned to users	Establish procedure for password assignment to a new user when enrolled.
C256	Parameter, Password	Password Required = yes	Set password parameter for assignment requires that a password exist
C278	Parameter, Password	Password Expiration = 120 days	Set maximum period for use of a password to 120 days
C294	Procedure	Password procedures on sharing	Establish procedures to forbid sharing passwords
C311	Penalties	Penalties for security violations	Establish personnel penalty for security violations
C412	Training	Password procedure training	Cover password procedures in security awareness training

Table 9 Task Definitions

TKcode	Type	Tasks	Description
TK071	Procedure	New Account Procedure	Write a detailed procedure establishing the requirements for establishing a new user account. Identify parameters
TK072	Procedure	System Password Parameters	Write a detailed procedure identifying the system password parameters.
TK132	Parameter	Implement System Password Parameters	Implement detailed procedures identifying system password parameters

Figure 3. Risk Assessment Example (page 2 of 3)

Table 10 Tests to Requirements

Rcode	TScode
R001	TS101
R002	TS103
R001	TS102

Table 11 Tests to Vulnerabilities

Vcode	TScode
V001	TS101
V002	TS107
V003	TS109

Table 12 Tests to Evidence

Ecode	TScode
E201	TS101
E202	TS102
E203	TS102

Table 13 Requirement/Vulnerability Status

Rcode/Vcode	Status	Ecode	Reason
V001	Open		
R001	Closed	E201	Password required on default setup and shows on all accounts.
R002	Closed	E205	The password expiration is reasonably set.

Table 14 Requirement/Threat/Risk Table

Vcode	Tcode	RKcode
R001	T10	RK039

Table 15 Vulnerability/Threat/Risk Table

Vcode	Tcode	RKcode
V001	T10	RK017

Table 16 Accepted Residual Risk List

RKcode	Manager	Reason
RK017	Site Security Officer	Procedures forbid sharing passwords; personnel penalty for violations; training courses cover.

Table 17 Risks to Countermeasures Table

RKcode	**Ccode**
RK001	C254
RK001	C255
RK001	C256

Table 18 Countermeasures to Tasks Table

Ccode	**TKcode**
C254	TK107
C255	TK109
C256	TK103

Figure 3. Risk Assessment Example (page 3 of 3)